**Objected – Oriented Programming**

**Drink Marketplace. Endterm project.**

**SE – 2211 | K. Alibi, Zh. Erulan, N. Beksultan, A. Damir**

**Astana IT University**

**Contents:**

**Project Description\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1**

**Database Description\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1**

**Pattern Description\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2**

**Branches Description\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3**

**How it works?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4**

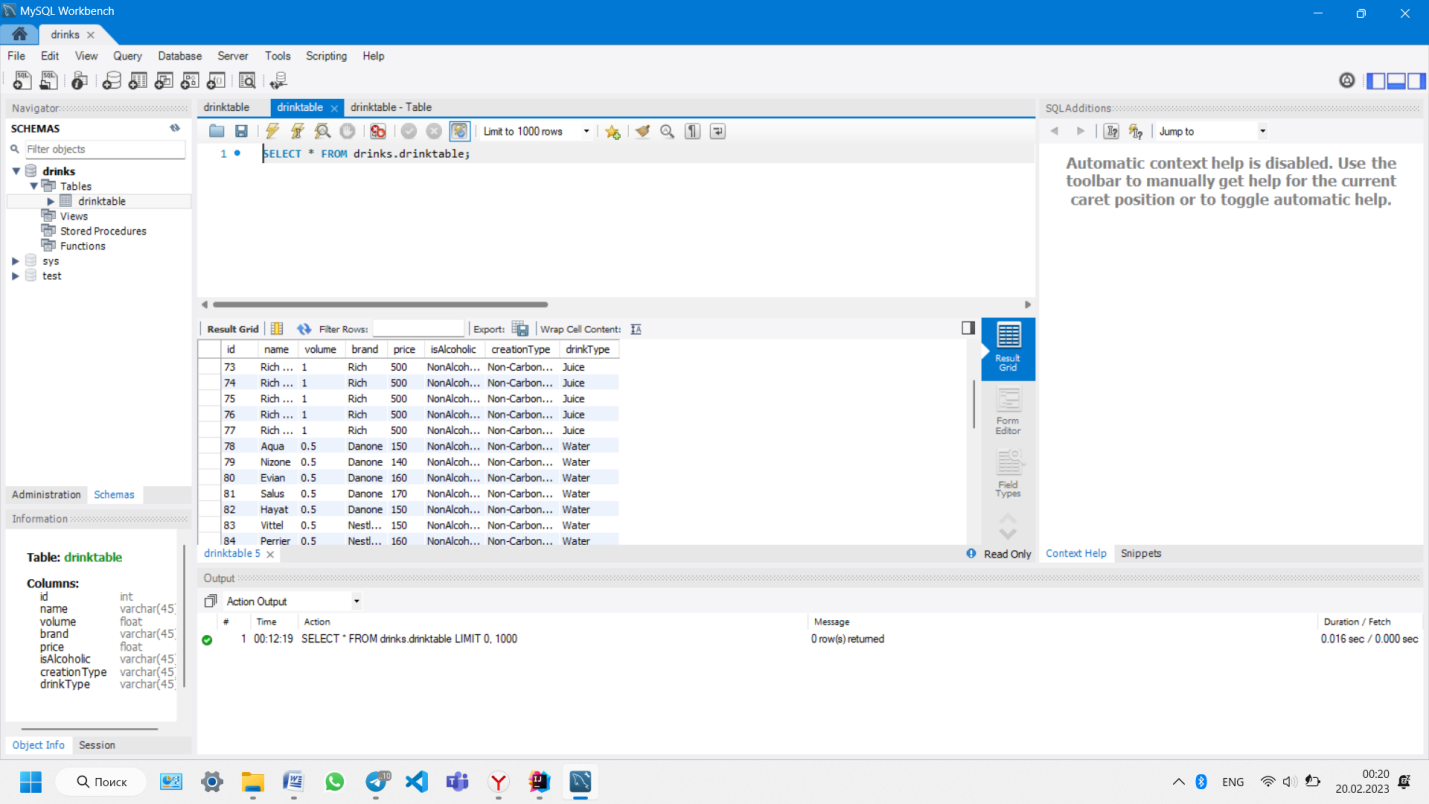
**Conclusion\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10**

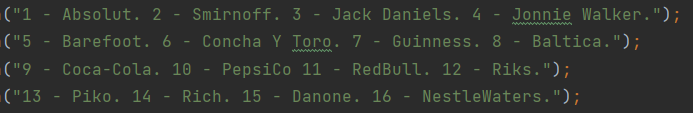
**Project Description:**

Our project is a program for the management of the beverage marketplace. Our program allows you to add a drink by entering its id, name, volume, brand and price. By adding a brand, the program automatically adds the properties of the drink. Determines whether the drink is alcoholic or not, if yes, then distilled or fermented, and if not, then carbonated or not. Next, he determines the type of drink. For example: vodka, wine, soda or plain water. All data adds in a table in a MySQL based database. After adding data, you can output this data to the console, change the data of the selected drink, delete specific or all data from the table at once, and you can also find the data you need using a search with a filter. In addition, I want to add that our program is written based on the Composite pattern.

**Database Description:**

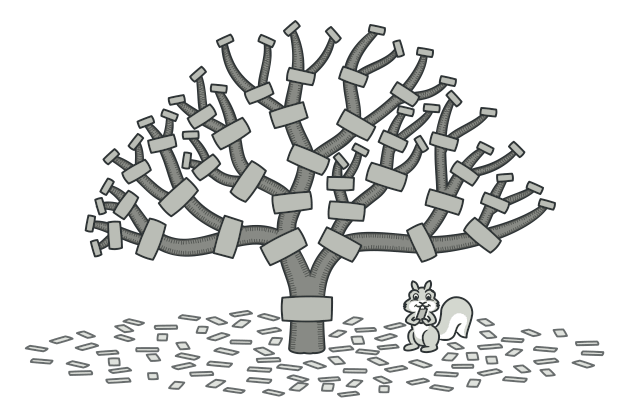
We chose MySQL as the database environment because it is more convenient and easy to learn. With the help of a special MySQL Workbench program, we created a database, and in it a table that contains columns such as: id, name of the drink, it’s volume, price, brand of this drink, whether the drink is alcoholic or not, the type of creation, and the type of drink itself. Creation type can include the following data: Distilled, Fermented, Carbonated and Non Carbonated. And the type of drink includes such data as: Whiskey, Wine, Vodka, Beer, Soda, Energy drink, etc. In addition, we added 16 different companies.

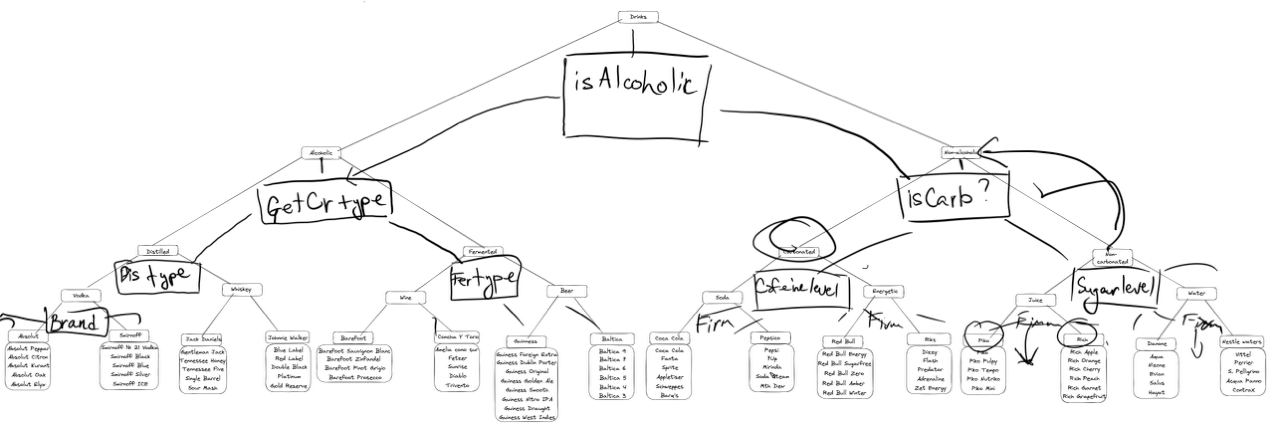




**Pattern Description:**

We used composite pattern. Composite pattern is a structural design pattern that lets you compose objects into tree structures and then work with these structures as if they were individual objects. Because we planned our project as binary-type hierarchy, composite pattern became our choice. Composite gave us easy to understand structure of your program and it became easier to treat individual classes individually and to compose objects uniformly.



****

**Branches Description:**

The drink category consists of two main categories: Non-alcoholic and Alcoholic drinks.

The Alcoholic drinks category is further divided into two sub – categories: Distilled and Fermented.

Under the Distilled category, there are two sub-categories: Vodka and Whiskey. Vodka includes two brands: Absolut and Smirnoff. Whiskey includes brands, like Jack Daniels and Johnnie Walker. These brands include a lot of different objects/drinks.

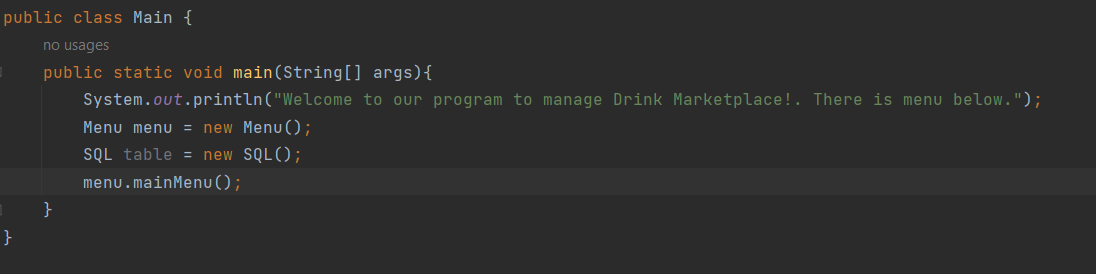
Under the Fermented category, there are two sub-categories: Wine and Beer. Wine includes Barefoot and Concha Y Toro brands. Beer includes Giunness and Baltica brands. These brands include a lot of different objects/drinks.

The Non-alcoholic drinks category is further divided into two sub-categories: Non-carbonated and Carbonated drinks.

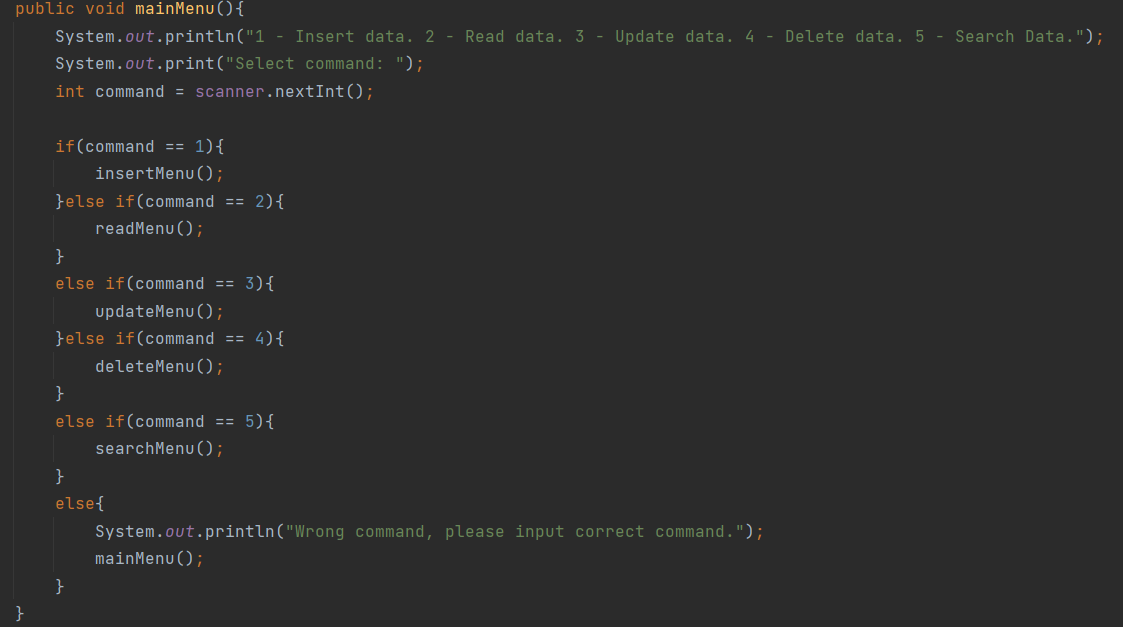
Under the Non-carbonated category, there are two sub-categories: Juice and Water. The Juice category consists of two sub-brands, Piko and Rich. The Piko brand consists of five sub-brands: Piko, Piko Pulpy, Piko Tempo, Piko Nutriko, and Piko Mini. The Rich brand consists of six sub-brands: Rich Apple, Rich Orange, Rich Cherry, Rich Peach, Rich Garnet, and Rich Grapefruit. The Water category consists of two sub-brands, Danone and Nestle Waters. The Danone brand consists of five sub-brands: Aqua, Nizone, Evian, Salus, and Hayat. The Nestle Waters brand consists of five sub-brands: Vittel, Perrier, S. Pelligrino, Aqua Panno, and Contrax.

Under the Carbonated category, there are two sub-categories: Soda and Energetic. The Soda category consists of two sub-brands, Coca-cola and Pepsico. The Coca-cola brand consists of six sub-brands: Coca-cola, Fanta, Sprite, Appletiser, Schweppes, and Barais. The Pepsico brand consists of five sub-brands: Pepsi, 7up, Mirinda, Soda-dream, and Mth-Dew. The Energetic category consists of two sub-brands, Red-Bull and Riks. The Red-Bull brand consists of five sub-brands: Red-Bull energy, Red-Bull Zero, Red-Bull Sugarfree, Red-Bull Amber, and Red-Bull Winter. The Riks brand consists of five sub-brands: Dizzy, Flash, Predator, Adrenaline, and Zet Energy.

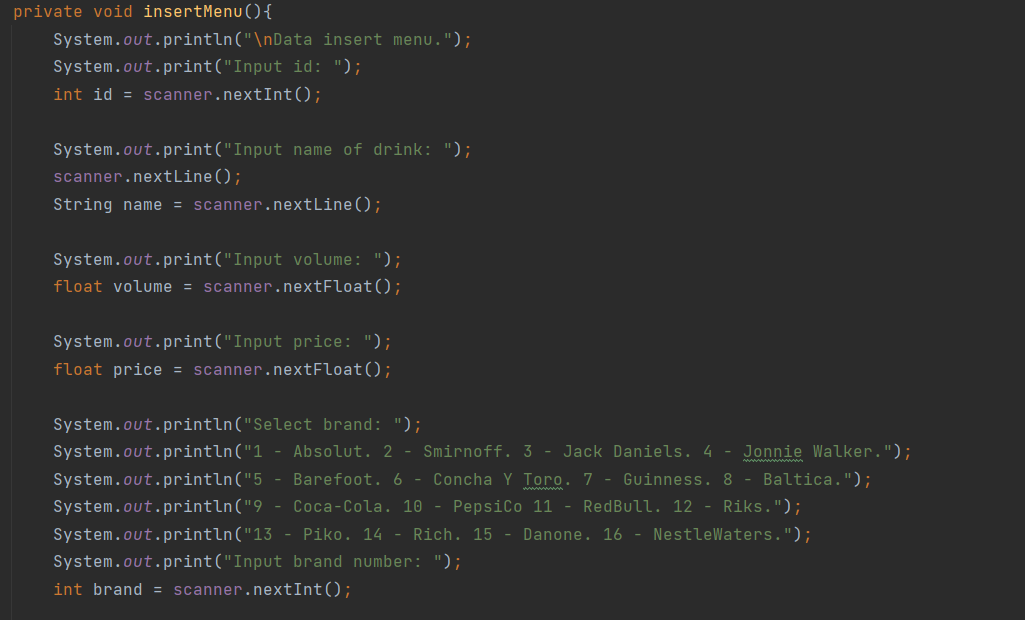
**How it works?**

At first program wil say hello to you. After that Main class will activate mainMenu method from Menu class. 

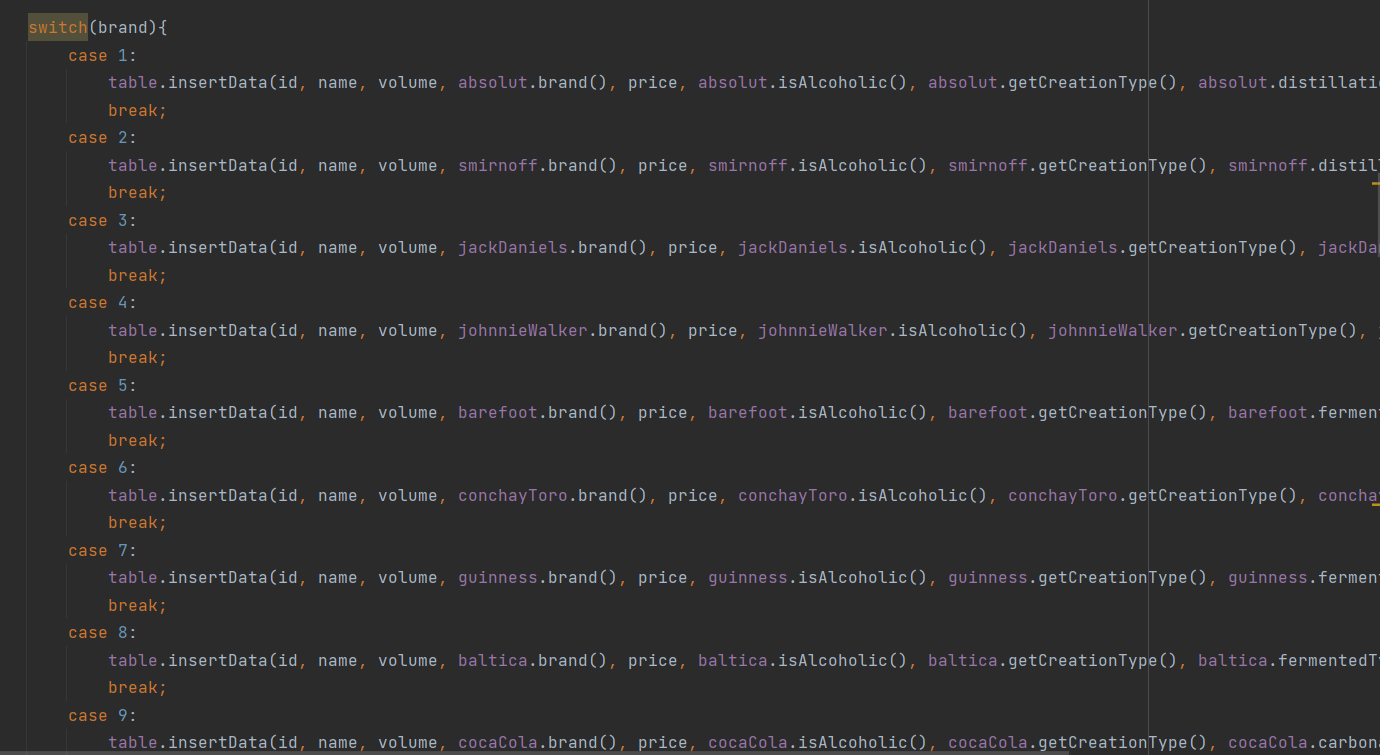
Menu class will give you to choose commands, like insert data, read data, update data, search data. After choosing command program will run methods that we choose.



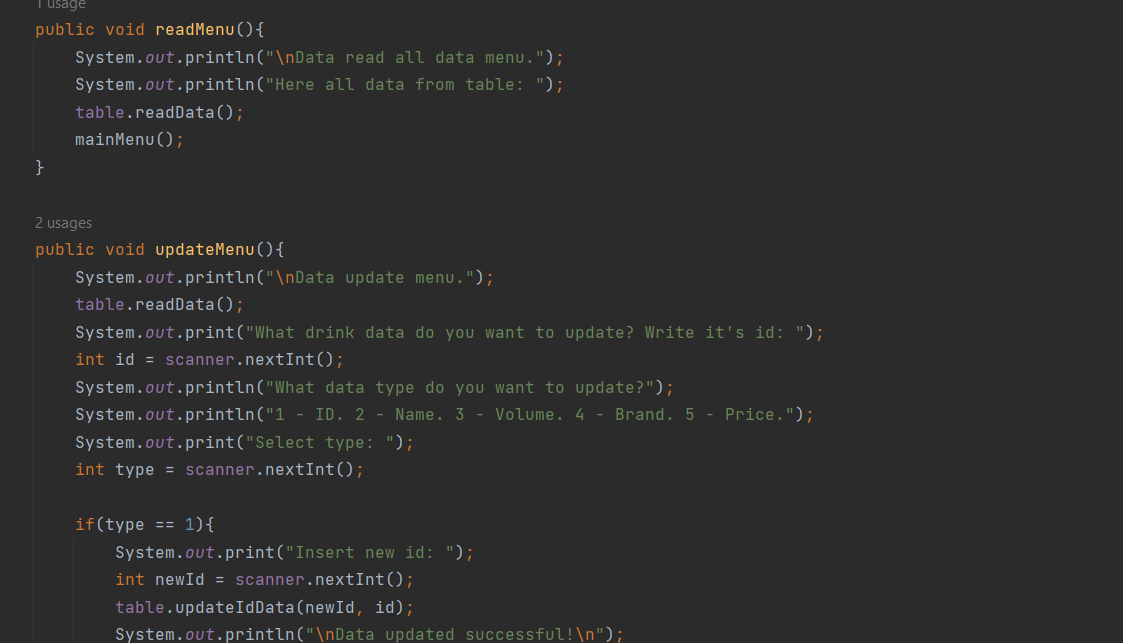
After choosing commands methods from Menu class will run methods from SQL class, but before it method will ask you to input data. For example, inputMenu method will ask you id, name, volume, brand and price to put it in table.

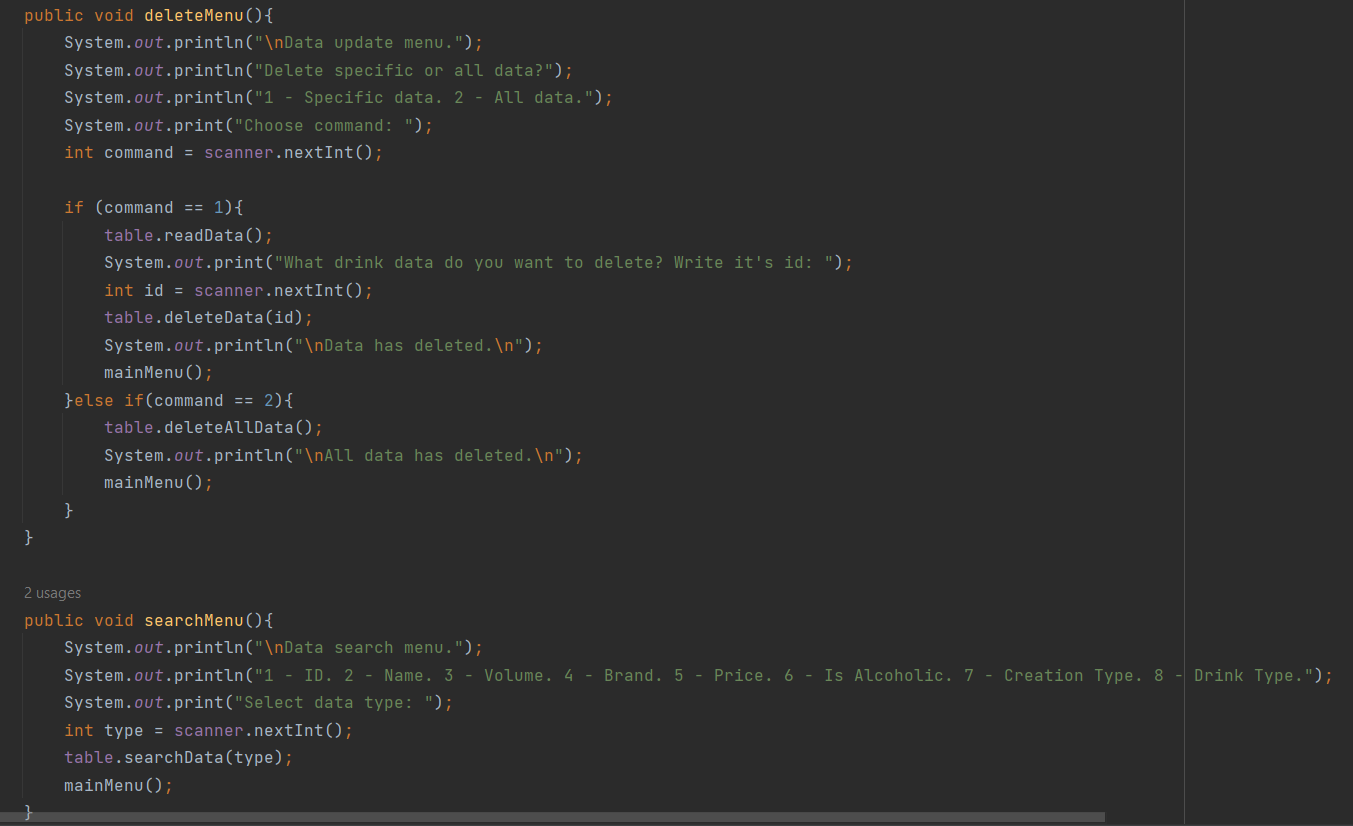


There are many switch cases for each brand. After choosing brand, program will put into table parameters like, brand, isAlcoholic, creationType and drinkType.

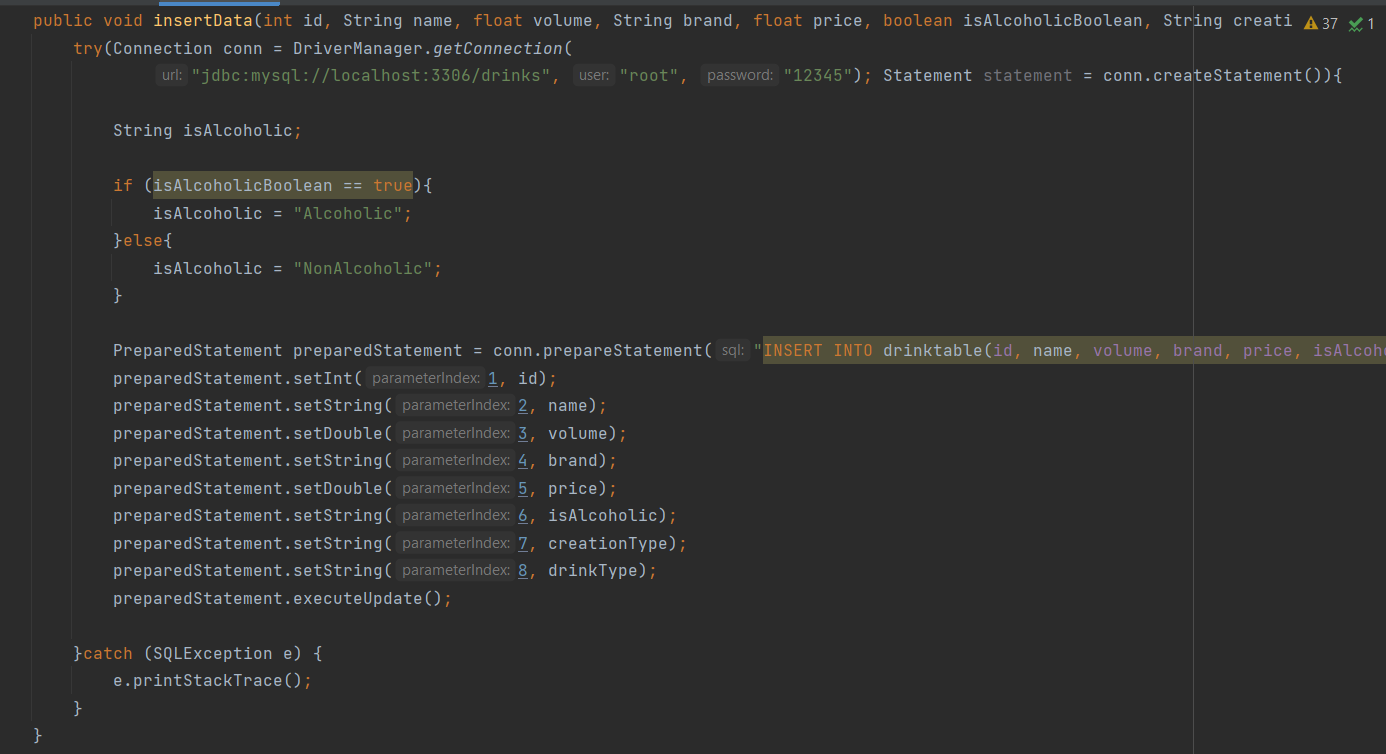


There are other commands in screenshots below.

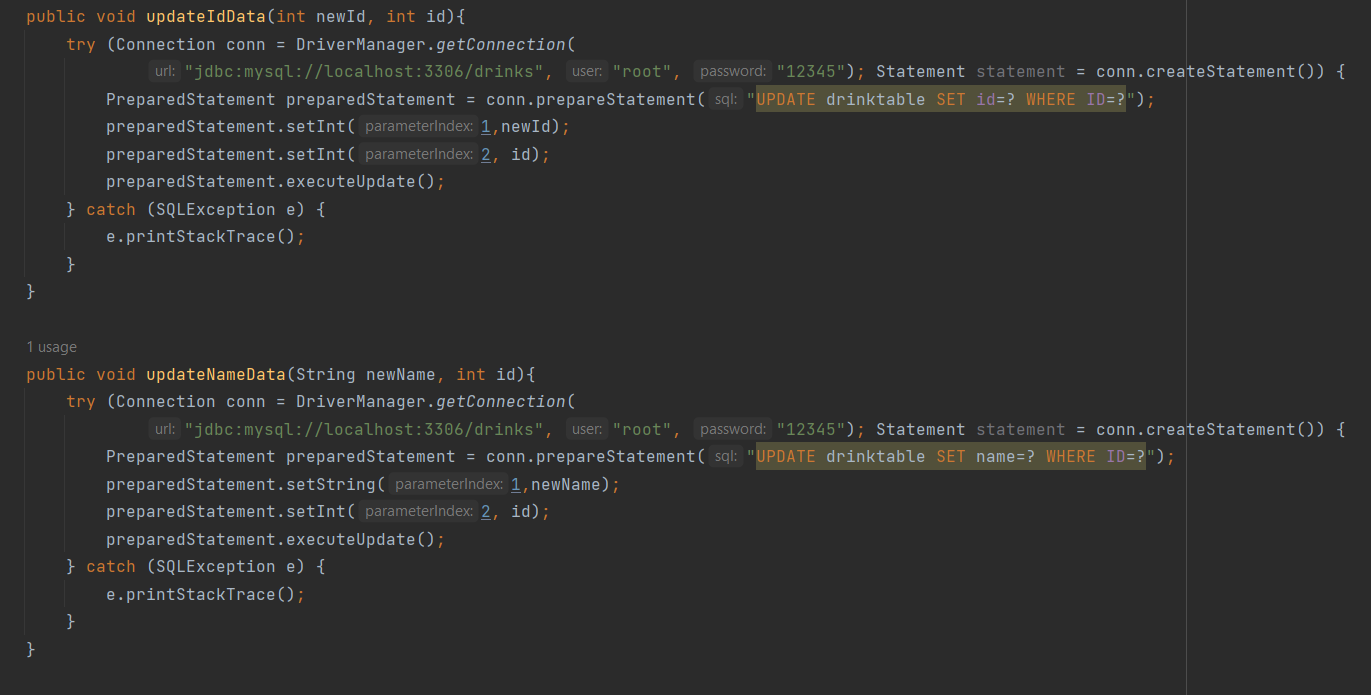




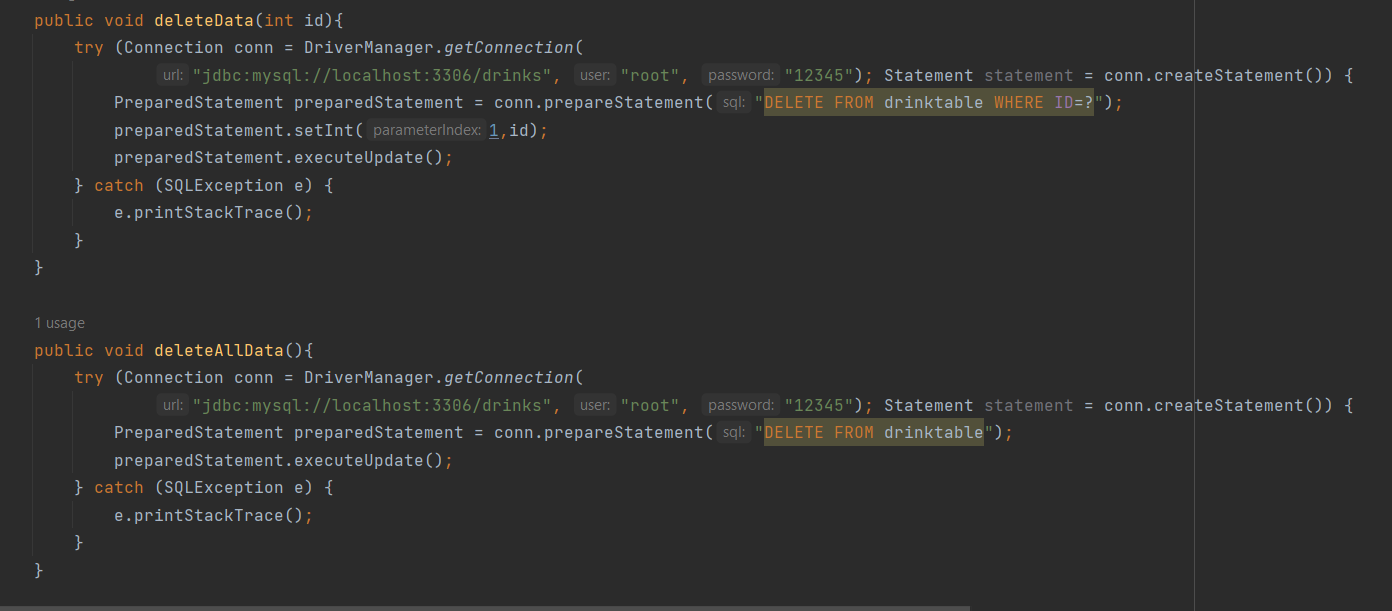
There are methods from SQL class in these screenshots. At first insertData method will insert data by user into database table.



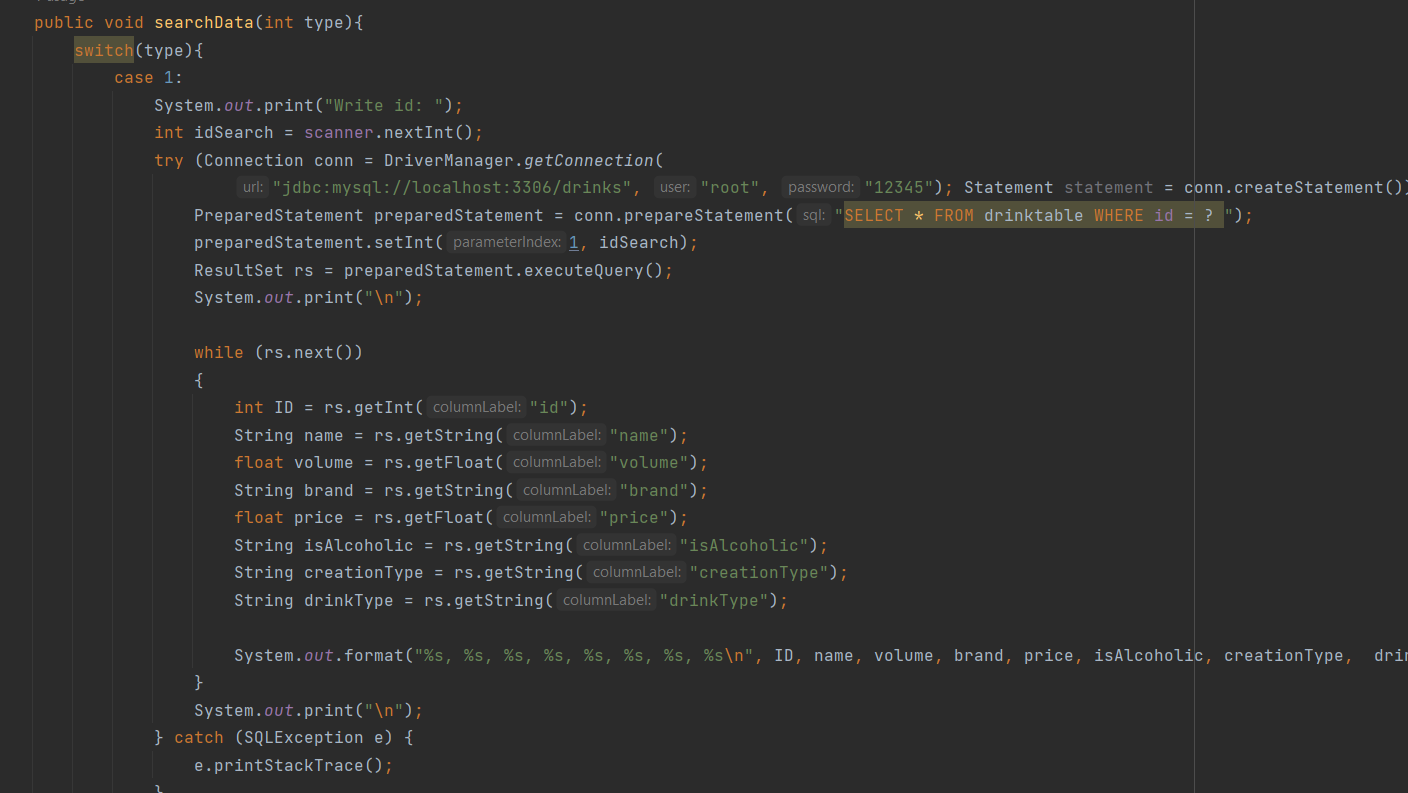
There is readData method which it will display table on console in this screenshot.

For updating data we have created some methods for each parameter of table, like updateIdData, updateNameData and etc. 

There are two methods: deleteData and deleteAllData. Method deleteData will delete data which user has chose. Method deleteAllData will delete all data in table.



The last method in SQL class is searchData method. This method will display data that user needs on console. It has some switch case for each parameter.





In our project, we have implemented the S.O.L.I.D. principles as follows:

S (Single Responsibility Principle) - Each class is responsible for a small part of our project's functionality. Classes do not execute excessive functionality and only implement necessary functionality.

O (Open-Closed Principle) - Classes in our project are closed for modification but open for extension.

L (Liskov Substitution Principle) - Our project has a five-step hierarchy with many superclasses and subclasses. Each superclass can be substituted by its subclasses because the subclasses implement all the interfaces of their superclass.

I (Interface Segregation Principle) - Classes do not implement unnecessary or excessive functionality.

D (Dependency Inversion Principle) - Classes in our project depend on abstract classes and interfaces rather than on other classes or functions.

**Conclusion:**

In conclusion, let's say that we have learned how to write a program with a 5-step hierarchy and database management. We also learned about patterns and were able to implement a project based on one.